The use of OpenSource technologies for distributing historic maps and creating search engines for searching though the catalogues

Manfred Buchroithner*,János Jeney*+**

* Technical University Dresden
** Eötvös Loránd University, Budapest

Abstract. As the preserving of historic maps are of great importance to map historians, it has often come into question of how they can be preserved using digital technologies. Today OpenSource software is developing at a fast pace, competing constantly with commercial products in many areas. Unfortunately spatial applications under OpenSource are still rare and if they do exist are unstable, lack features or are very difficult to use. On the other hand more than fifty per cent of the world webservers run Linux with an OpenSource web server and database server package. Although these are not designed for spatial features it is possible to utilize them for this purpose. Spreading historic maps as raster images is possible using these technologies, as well as with the use of SVG it is possible to create interactive thematic maps. The latter can be used to analyze these old maps, as well as for creating user interfaces for search engines to search in map catalouges.

Keywords: OpenSource, SVG, Web Mapping

Introduction

As the preserving of historic maps are of great importance to map historians, it has often come into question of how they can be preserved using digital technologies. Today OpenSource software is developing at a fast pace, competing constantly with commercial products in many areas. Quite often skilled users and programmers who download the software from the Internet fix

errors and bugs much faster than a software company does, since quite often the developers are themselves then users. These software packages are becoming more and more user-friendly, targeting the average user who has little or no knowledge of programming. Unfortunately the OpenSource spatial applications are rare and if they do exist are unstable, lack features or are very difficult to use. Since the OpenSource software is not documented as well as the commercial software, the support comes from the users. In many cases where there are many users, this type of support can be better then a commercial support network, but if there are few users this is a poor source of information.

On the other hand more than fifty per cent of the world webservers run Linux, which is a Unix-like, stable OpenSource operating system. They usually run apache, which is also an OpenSource application. In many cases they use a background database running on PostgresSQL, or in some cases MySQL, which are also OpenSource software packages. Unfortunately these give us very limited GIS features. On the other hand it is very useful in spreading pictures on the Wold Wide Web.

Displaying raster images on the World Wide Web

Since scanned historic maps are raster images, they can be displayed by any web browser. There are many JavaScript applications that let the user pan and zoom these images within a web page. Although this does not make use of any spatial features, it gives us an opportunity to display the map. These images of the maps can be entered into a database serving the background of the webpage and many search methods can be made. Once the search is completed, the user can view the map and pan and zoom it. It is important that bandwidth is utilised as well as possible. It is important that these images get downloaded over the network in their full size that the server sends them, so reducing the size on the client side can use up much bandwidth. On the other hand if the image is reduced on the server side, or if presized images are prepared on the web server, the client will have to download the larger image if he wants to zoom in. As can be seen, both extremes can be bad, so an optimization has to be made to reduce network traffic as much as possible.

In my example I had two preview sizes on the server, one for the previews in the search result and another for a slightly larger preview if the user selects a map. If the user wants to pan and zoom the map, the full size image gets downloaded.

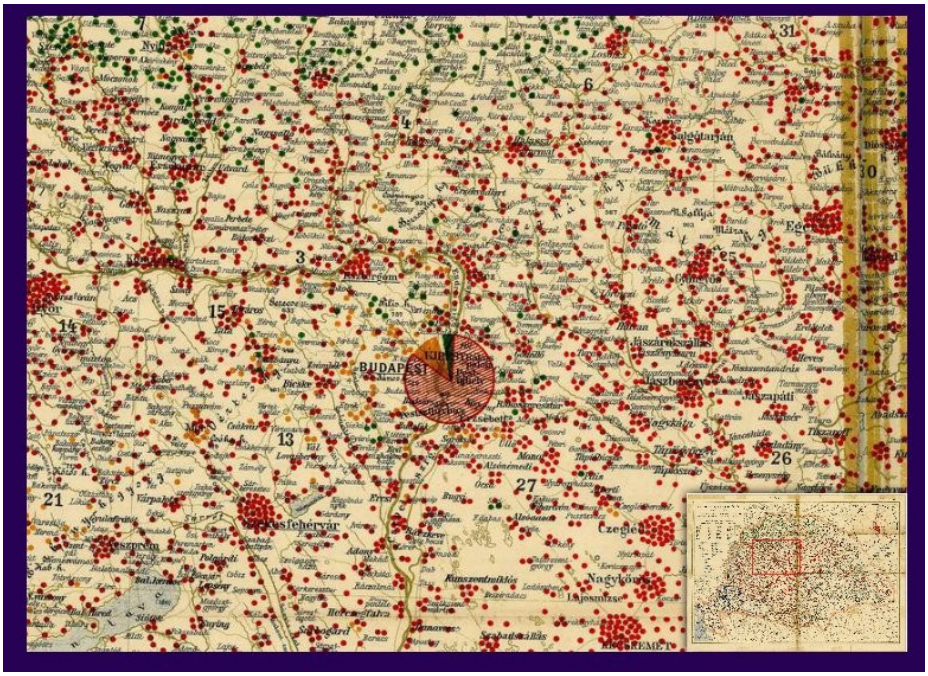

Image 1: A raster map loaded using shiftzoom. the user can pan, zoom or use the overview using the mouse.

Using SVG to create interactive maps on the World-Wide Web

SVG (Scalable Vector Graphics) is the standard format for displaying vector graphics on the world wide web. It is widely used to display static images. Its features on the other hand go far beyond this. The file itself is an XML format text file with tags and features. Each tag has an ID, styles, features and events, which can either be static, in which case they are coded in the XML file itself, or they can be added or modified interactively by a script run by the web browser. this means that a script running on

the client side can change features and styles for lines or polygons for example colour, border width etc..., while the vector image is displayed on the users browser. Each object on the vector image is one tag in the XML file. If this image is a map, the objects are lines and polygons. Point features are either small circles or simple polygons and have to be dealt with by the browser as such.

Using AJAX technology a dataset can be downloaded to the client. The features can be modified using this dataset. using this method, we can modify the features of the individual elements on the vector map on the basis of the results of a query in a database. Each object gets its features modified by its ID, so it is important, that the ID is entered in the SVG file. The ID is the only feature of the tags that cannot be modified.

Using events, one can run predefined functions using JavaScript, including functions that are included in the web page containing the image, or even functions included in the image itself. These functions can perform anything that can be preformed by the web browser at the time of running, including, but not limited to modifying the styles and features on the image or on another image displayed on the web page. The event can also trigger functions that modify styles, features and events on the object that triggered the event. Events can also be added or removed by a script running on the browser.

2.1 Using SVG to create a search engine in a database containing historic maps

In the next example I made a search engine for searching in a database containing ethnic maps of the pre-world war I territory of Hungary. An SVG map with the counties of Hungary in 1914 was used as the user interface. Each polygon presents one county and has a unique id. The ethnic maps (raster images) in the database each have an entry regarding each county whether the territory of each county is shown, in whole or partially on the map or not. In the browser there is a variable for each county, which stores the search criteria regarding that particular county (ie: if the result should only contain maps that show that particular county in whole or in part, or if it should only contain maps that do not show that county at all). Each time the user clicks on the county, this value changes according to a predefined cycle along

with the colour of the county on the SVG map. The user can see by checking the legend of the map what the status of a particular county is. When the search runs the values of the variables of the counties, they are compared with the values in the tables and entries are displayed only where the values match. If the value of a variable of a county suggests that it is not a search criterion, then that value does not get checked. To avoid empty results, every time the user clicks on an object on the map, a quick search in the database checks how many results would come out using the given search criteria and warns the user if there are no results.

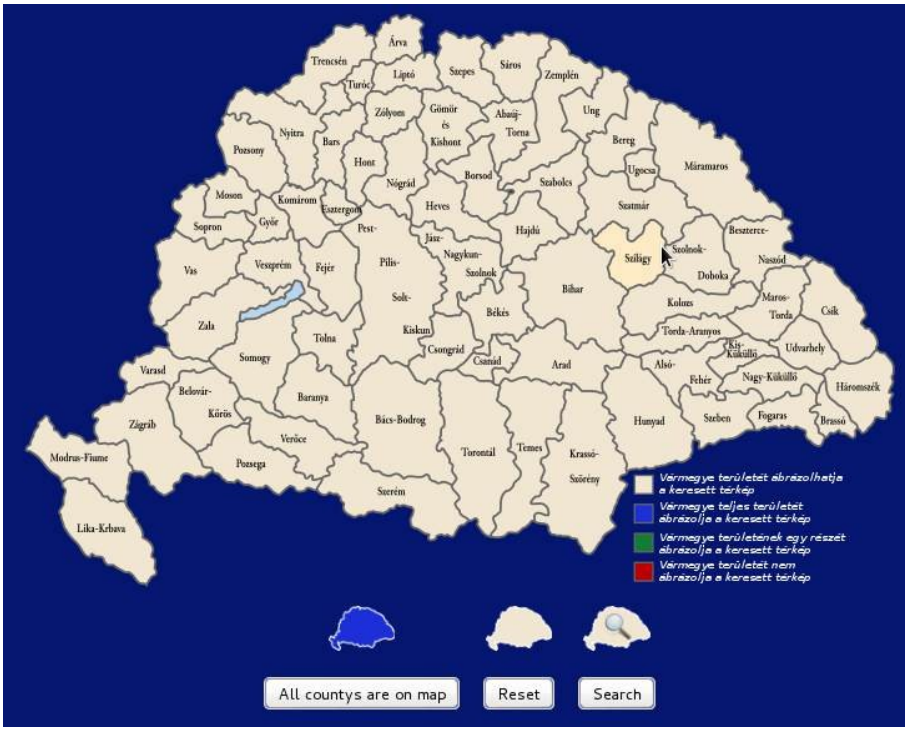

Image 2: Search engine using an SVG map of the counties of Hungary from 1914

2.2 Using SVG to analyze the maps in the database

Using the same database we can create an interactive map showing us which counties are shown on the map being viewed. The user can use either the above search method, or any other search method offered (eg. by Author, title or by selecting the map from a list) and can get an SVG map generated showing what counties are on the map. We use the same database as in the above example. Depending on the value in the database entry associated with the raster map concerned regarding the fact of a particular county is shown on the map or not, the script assigns the county a colour. In this way a map appears according to a predefined legend that shows us what counties are In this way a map appears according to a predefined legend that shows us what counties are shown on the map being viewed, what counties are not, and what counties are shown only partially.

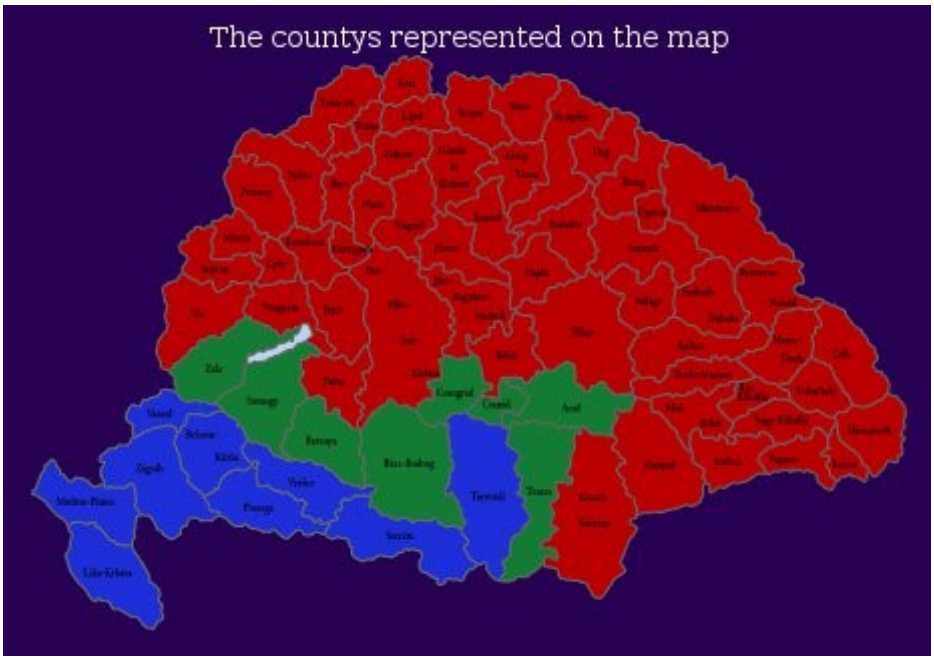

Image 3: A map of Hungary showing the counties as in 1914. The counties that are shown the the ethnic map by Sir Arthur Evens are blue, the counties on partially on the map are green, the ones not on the map are red

2.3 Using SVG to analyze the full catalogue

If for example we want to see how many maps in the catalogue show the different counties, we can create a thematic map. A query counts how many entries there are in the entire database that show a particular county. This query is checked for every county, and the polygon of the county gets filled with a colour according to a predefined legend depending on the result from the count. This query can also be done for part of the catalogue, for example only for maps that are in a particular archive. Putting this result regarding an archive in Germany next to a map showing the German population according to counties, one can see a slight parallel in what areas are shown mostly on the maps found in Germany, thus making this tool useful for analyzing the full catalogue of the archive.

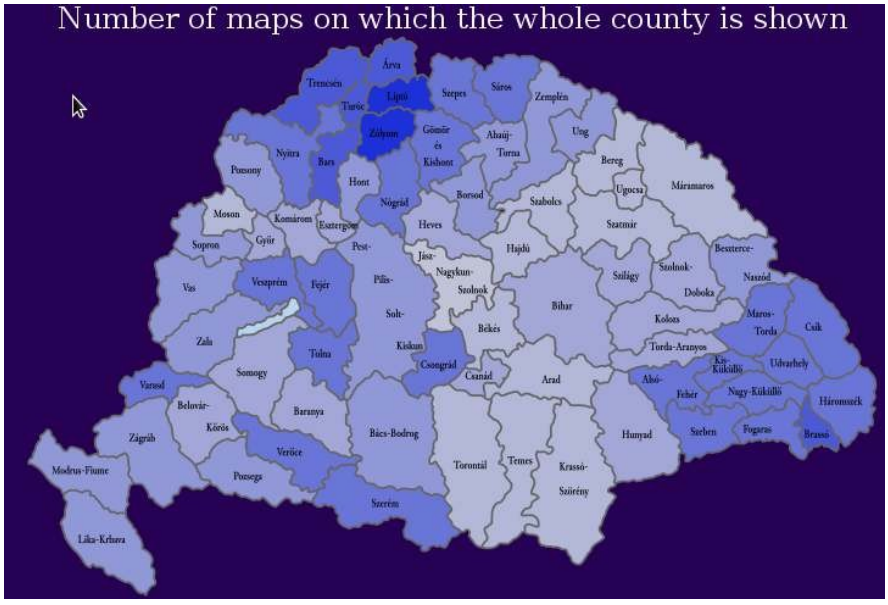

Image 4: Thematic SVG map of Hungary in 1914 showing how many maps in the catalogue show each county. Thematic data applied by the browser.

2.4 Interactive topnymy on SVG maps

In the next example a map of Europe was made to allow searches in the database by selecting the actual place where the original map is currently stored on a map. Since the web page is in three languages the names of the countries have to be written in the language the web page in viewed in. In order not to make three maps the names of the countries with unique IDs were written in three languages (Hungarian, German, English) in a table in the database. An empty text element was created on top of the country's polygon and the contents added interactively by the browser from the database according to the language selected by the user. If the language gets changed the names on the map get changed accordingly.

Image 5: Map of Europe with English toponymy

Image 6: Map of Europe with Hungarian toponymy

Conclusion

Although these tools are not designed for mapping applications, they are very well utilized for certain limited mapping services of displaying raster images and creating simple thematic maps.

References

Shiftzoom online documentation
http://www.netzgesta.de/shiftzoom/ Accessed 12 November 2012

w3schools.com SVG tutorial

http://www.w3schools.com/svg/svg_inhtml.asp Accessed 12 November 2012

w3schools.com JavaScript DOM tutorial

http://www.w3schools.com/js/js_htmldom.asp Accessed 12 November 2012

Hungarian Ethnic Maps, Department of Cartography, Eötvös Loránd University

http://demogmap.elte.hu Accessed 12 November 2012